

Create Table

Most commercial DBMS use the relational model. Under the relational model, all data stored in the database is kept in tables. A table is like a two-dimensional array and consists of rows and columns. Each column records one attribute while each row records the values for all attributes that describe one instance of data. For example, we could use a table to record information about Invoices. If we needed to record the Invoice Number, Invoice Date, the Number of the Customer whom the invoice is for and the Total Amount Owning for the Invoice we would need four columns in the table. Each row in the table would record the information for one specific invoice.

Invoices Table:

Invoice Number	Invoice Date	Customer Number	Total
100	Jan 10, 2017	7	100.00
101	Jan 11, 2017	20	75.00

The Create Table statement defines a table object in the database. The statement specifies the name of the table and information about each column in the table. This includes:

- The name of the column
- The data type of the column
- Whether null is an acceptable value for the column
- Constraints that must hold true for the column (Constraints will be dealt with later)

Table names and column names should be descriptive. Limit the use of abbreviations in table or column names if possible. The bottom line is that a user of the database (e.g. a programmer) should be clear about what data is stored in a table or column based on its name. Some abbreviations are commonplace within industry (e.g. ID for Identification) and are acceptable.

Experience will guide you as to what is an acceptable name. Many organizations have standards that define what an acceptable name is. If standards are in place they must be followed. If a name consists of multiple words, capitalize the first letter in each word.

The name for a column that records the date of the invoice could be InvoiceDate or Invoice_Date depending on the company standards.

The name of an associative table (implements an associative entity) could include the names of the parent tables. For example: if Order and Item are the parent tables, then OrderItem or Order_Item could be the associative table name.

There is no default data type. You must explicitly state the datatype you want to use for each column.

The simplified syntax for the Create Table statement is shown below:

Create Table [If Not Exists] ***table_name***

```
(  
    {  
        column_name data_type [column_constraint [ ... ] ]  
        |  
        table_constraint  
    }  
    [, ... ]  
);
```

column_constraint

[Constraint ***constraint_name***]

```
{  
    Not Null |  
    Null |  
    Check (expression) |  
    Default default_expression |  
    Unique |  
    Primary Key |  
    References ref_table [ (ref_column) ]  
}
```

table_constraint

[Constraint ***constraint_name***]

```
{  
    Check (expression) |  
    Not Null (column_name) |  
    Unique (column_name [, ... ] ) |  
    Primary Key (column_name [, ... ] ) |  
    Foreign Key (column_name [, ... ] ) References ref_table  
    [ (ref_column [, ... ] ) ]  
}
```

Constraint name is an optional name for a column or table constraint. If the constraint is violated, the constraint name is present in error messages, so constraint names like “column must be positive” can be used to communicate helpful constraint information to client applications. Double quotes are needed to specify constraint names that contain spaces. If a constraint name is not specified, the system generates a name. In SDEV1201, constraint names are mandatory! Constraints will be discussed in a future lesson and document.

Datatypes (Simplified)

Numeric Types

Name	Storage Size	Description	Range
smallint	2 bytes	small-range integer	-32768 to +32767
integer	4 bytes	typical choice for integer	-2147483648 to +2147483647
bigint	8 bytes	large-range integer	-9223372036854775808 to +9223372036854775807
decimal	variable	user-specified precision, exact	up to 131072 digits before the decimal point; up to 16383 digits after the decimal point
money	8 bytes	currency amount	-92233720368547758.08 to +92233720368547758.07

An integer is a whole number (i.e. no decimal places). Integers can be positive, negative, or zero.

Decimal (*precision, scale*)

The *precision* of a decimal datatype is the total number of significant digits (i.e. not leading zeroes, commas or the decimal point) in the complete

number, that is, the number of digits to both sides of the decimal point. The *scale* of a decimal datatype is the number of decimal digits in the fractional part (i.e. to the right of the decimal point). Decimal and numeric are equivalent datatypes.

The money datatype stores a currency amount with a fixed fractional precision which is determined by the *lc_monetary* parameter based on the database's *Locale* setting. *Locale* refers to an application respecting cultural preferences regarding alphabets, sorting, number formatting, etc. In North America, money output would be a dollar sign, at least one digit to the left of the decimal point, and two digits to the right of the decimal point. If the value is over 999, commas will delimit thousands/millions/billions, etc.

Character Types

char (*n*)

varchar (*n*)

If specified, the length *n* must be greater than zero and cannot exceed 10,485,760. If varchar is used without length specifier, the type accepts strings of any length. If char lacks a specifier, it is equivalent to char (1).

Values of type char are physically padded with spaces to the specified width *n* and are stored and displayed that way. However, trailing spaces are treated as semantically insignificant and disregarded when comparing two values of type char.

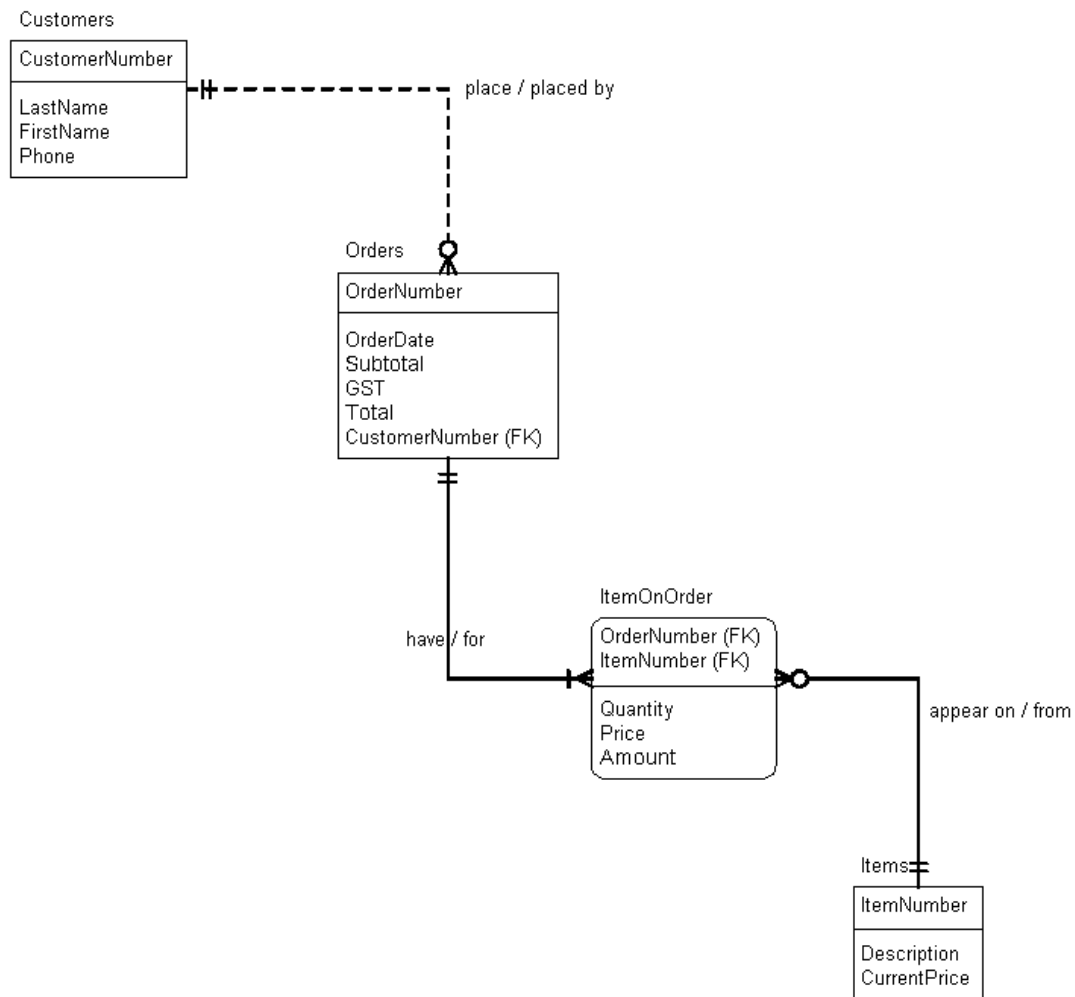
Date/Time Types

Name	Storage Size	Description	Low Value	High Value	Resolution
timestamp [(<i>p</i>)] [without time zone]	8 bytes	both date and time (no time zone)	4713 BC	294276 AD	1 microsecond
timestamp [(<i>p</i>)]	8 bytes	both date and	4713	294276	1

Name	Storage Size	Description	Low Value	High Value	Resolution
with time zone		time, with time zone	BC	AD	microsecond
date	4 bytes	date (no time of day)	4713 BC	5874897 AD	1 day

The time portion of all Date columns is automatically set to midnight (24-hour clock: 00:00:00, 12-hour clock: 12:00:00).

Possible code for creating tables for the Order example, are shown below.



Create Table Customers

```
(
    CustomerNumber    integer,
    LastName           varchar (50),
    FirstName          varchar (50),
    Phone              char (10)
);
```

Create Table Orders

```
(
    OrderNumber        integer,
    OrderDate          date,
    CustomerNumber     integer,
    Subtotal           decimal (7,2),
    GST                decimal (5,2),
    Total              decimal (9,2)
);
```

Create Table Items

```
(
    ItemNumber         integer,
    Description         varchar (100),
    CurrentPrice        decimal (6,2)
);
```

Create Table ItemOnOrder

```
(
    OrderNumber        integer,
    ItemNumber          integer,
    Quantity           integer,
    Price              decimal (6,2),
    Amount             decimal (7,2)
);
```